

WHAT IS CLAIMED IS:

1 1. A method of prioritizing data transfer requests
2 serviced by a centralized data transfer unit comprising the
3 steps of:

4 receiving transfer request packets, each transfer request
5 packet indicating a desired data transfer and a priority level
6 within a hierarchy of a plurality of priority levels;

7 storing each received transfer request packet in a queue
8 memory, each received transfer request packet stored in a
9 first-in-first-out fashion within each priority level;

10 detecting availability of free data transfer channel
11 among a plurality of data transfer channels within said
12 centralized data transfer processor; and

13 dispatching a next transfer request packet in said first-
14 in-first-out fashion to the corresponding free data transfer
15 channel.

1 2. The method of claim 1, wherein:

2 said step of storing each received transfer request
3 packet in a queue memory includes

4 storing each received transfer request packet in a
5 random access memory fashion,

6 defining an address range within said queue memory
7 allocated to each priority level with a corresponding
8 queue bounds register,

9 storing a next input transfer request packet for a
10 priority level at address within said queue memory
11 indicated by a corresponding queue write pointer
12 indicating address location, and

13 reading a next output transfer request packet for a
14 priority level from an address within said queue memory
15 indicated by a corresponding queue read pointer.

1 3. The method of claim 2, further comprising the steps
2 of:

3 incrementing the corresponding queue write pointer upon
4 storage of a transfer request packet in queue memory; memory
5 and

6 decrementing the corresponding queue read pointer upon
7 transfer of a transfer request packet to a data transfer
8 channel.

1 4. The method of claim 2, further comprising the steps
2 of:

3 dynamically defining said address range allocated to a
4 priority level by dynamically writing to a corresponding queue
5 bounds register.

1 5. The method of claim 1, further comprising the steps
2 of:

3 detecting if a channel corresponding to a priority level
4 within the queue is empty; and

5 if a received transfer request packet of has a priority
6 level detected to be empty

7 bypassing storing said transfer request packet in
8 said queue memory, and

9 dispatching said transfer request packet directly
10 to the corresponding free data transfer channel.

1 6. The method of claim 1, further comprising the steps
2 of:

3 generating said transfer request packets at each of a
4 plurality of transfer request nodes; and

5 upon dispatching of a transfer request packet to a free
6 data transfer channel, sending a queue acknowledge signal to
7 said transfer request node originating said transfer request
8 packet.

1 7. The method of claim 1, further comprising the steps
2 of:

3 generating said transfer request packets at each of a
4 plurality of transfer request nodes, said transfer request
5 packet identifying said originating transfer request node;

6 disabling dispatching transfer request packets
7 originating from selected transfer request nodes; and

8 ignoring transfer request packets originating from
9 disabled transfer request nodes.

1 8. The method of claim 8, wherein:

2 said step of disabling dispatching transfer request
3 packets originating from selected transfer request nodes
4 includes writing at least one to a corresponding location
5 within a request disable register.

1 9. The method of claim 9, wherein:

2 the number of data transfer channels equals the number of
3 priority levels.